General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some
 of the material. However, it is the best reproduction available from the original
 submission.

Produced by the NASA Center for Aerospace Information (CASI)

Agristars

'Made available under NASA sponsorship in the interest of early and wide dissemination of Earth Resources Survey Program information and without liability for any use made thereot."

Supporting Research

SR-X1-04033 NAS9-15981

E83-10309

A Joint Program for Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing

January 1981

ASSEMBLY LANGUAGE CODING FOR CLASSY

(E83-10309) ASSEMBLY LANGUAGE CODING FOR CLASSY (Elogic, Inc.) 36 p HC A03/MF A01 CSIL 02C

N83-27299

Unclas G3/43 00309

M. E. Rassbach



Elogic, Inc. 4242 S.W. Freeway, Suite 304 Houston, Texas 77027



NASA







Lyndon B. Johnson Space Center Houston, Texas 77058

A. Title and Subtite Assembly Language Coding for CLASSY 7. Author(s) 11. E. Rassbach 9. Performing Organization Name and Address Elogic, Inc. 4242 S.M. Freeway, Suite 304 Houston, TX 77027 12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, TX 77058 15. Supplementary Notes 16. Abetract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) Unclassified 20. Security Classif. (of this page) 21. No. of Pages 22. Price* 19. Security Classif. (of this page) Unclassified 22. Price* 23. Price*	1, Report No. SR-X1-04033	2, Government Acces	sion No.	3. Recipient's Catalog	g No.
7. Author(s) 11. E. Rassbach 12. Performing Organization Name and Address Elogic, Inc. 4242 S. M. Freeway, Suite 304 Houston, TX 77027 12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, TX 77058 15. Supplementary Notes 16. Abetract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	4. Title and Subtitle			5. Report Date	
### H. E. Rassbach 9. Performing Organization Name and Address Elogic, Inc. 4242 S.M. Freeway, Suite 304 Houston, TX 77077 12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, TX 77058 15. Supplementary Notes 16. Abstract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Supplementary Notes) 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	Assembly Language Coding fo	r CLASSY		6. Performing Organi	zation Code
Elogic, Inc. 4242 S.M. Freeway, Suite 304 Houston, TX 77027 12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, TX 77058 15. Supplementary Notes 16. Abstract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*				NAS811	ration Report No.
4242 S.M. Freeway, Suite 304 Houston, TX 77027 12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, TX 77058 15. Supplementary Notes 16. Abstract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Supplementary Notes) 18. Distribution Statement 19. Security Classif, (of this report) 20. Security Classif, (of this page) 21. No. of Pages 22. Price*	9. Performing Organization Name and Address			10, Work Unit No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, TX 77058	4242 S.W. Freeway, Suite 30	4		NAS9-15981	
Lyndon B. Johnson Space Center Houston, TX 77058 15. Supplementary Notes 16. Abetract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement CLASSY algorithm 19. Security Classif. (of this page) 21. No. of Pages 22. Price*					
15. Supplementary Notes 16. Abstract A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement CLASSY algorithm 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	Lyndon B. Johnson Space Cen		ion	14. Sponsoring Agency	/ Code
A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*			 		
A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
A set of assembly language improvements to the CLASSY clustering algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
algorithm have been developed and tested. These are descrived in detail and analyzed. 17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	16. Abstract				
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	algorithm have been develop				
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	•				
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
17. Key Words (Suggested by Author(s)) CLASSY algorithm 18. Distribution Statement 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	·				
CLASSY algorithm 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					•
CLASSY algorithm 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
CLASSY algorithm 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*					
CLASSY algorithm 19. Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	17. Key Words (Suggested by Author(s))		18. Distribution Statement	· · · · · · · · · · · · · · · · · · ·	
19, Security Classif. (of this report) 20. Security Classif. (of this page) 21. No. of Pages 22. Price*	CLASSV algorithm				
	CEASST argurreriii				
	19. Security Classif. (of this report)	20. Security Classif. (c	of this page)	21. No. of Pages	22. Price*
	Unclassified	Unclassifie	ed	35	

ASSEMBLY LANGUAGE CODING FOR CLASSY

BY

M. E. RASSBACH

This report describes Classification activities of the Supporting Research project of the AgRISTARS program.

Elogic, inc. 4242 S.W. Freeway, Suite 304 Houston, Texas, 77027

CONTENTS

Purpose	1
Matrix Arithmetic	2
New Exponential Routine	5
Path Difference in CLASSY	9
Appendix 1: Program listings for vector routines .	11
Appendix 2: Program listings for the fast exponential system	26

Assembly language coding for CLASSY

Elogic has developed several assembly language routines to speed up the CLASSY algorithm.

These fall into two catagories

- 1) matrix arithmetic
- 2) improvement in the exponential routine

Purpose of assembly-coded Matrix Arithmetic and Exponential Routines.

Currently available compilers produce a great number of extra instructions, particularly "overhead" instructions such as Load, Store, and index arithmetic. These instructions can be eliminated by careful hand coding of the routines. Thus it is possible to reduce the machine-processing time for a program by coding the most frequently used routines in assembly language.

CLASSY is particularly suited for this form of optimization, since most of its execution time is spent in a few vector and matrix processing routines. In addition, as was noted while designing the vector processing routines, application of the distributive law will allow putting one multiplication outside of the inner loop in several instances. (It was later realized that this change could be made in the Fortran version, as well.) Thus by using new, assembly language versions of the matrix arithmetic routines, CLASSY could be greatly accelerated. Some results are given in Table 1.

Matrix Arithmetic

The matrix arithmetic routines correspond exactly (bit-bybit) to Fortran versions ("new Fortran") of the old matrix routines. The new Fortran versions differ from the original Fortran versions ("old Fortran") by the application of the distributive law and precalculation of some quantities to reduce the number of operations inside the inner loop. Rounding error differences can make the new version differ from the old version, which may become substantial due to the characteristics of the CLASSY algorithm (See "Path Differences in CLASSY" below). However, since these differences are only due to floating-point rounding error, which is random in both the new and old versions, the results are equivalent. The new version may in fact give slightly higher precision, due to the method of summation used. Table 2 gives the routine names for the new and old Fortran versions, as well as the assembly version. The names of EXEC'S used to switch from one version to another are also given. names given are the names of the source language files; the object programs must have the original name.

Version	Routines (Vector, XP)	Iteration: Channels	Time s, (Improve- ment)	Time per Cluster per Iteration (Improvement)
Original	Old Fort., Fort. XP	10 4	369	4.10
Old Fort.	Old Fort., ALC XP	10 4	518 (0.71)	3.99 (1.03)
New Fort.	New Fort., ALC XP	10 4	301 (1.23)	3.34 (1.23)
ALC	ALC, ALC XP	10 4	240 (1.54)	2.67 (1.54)
Original	Old Fort., Fort XP	7 8	1590	15.14
Old Fort.	Old Fort., ALC XP	7 8	1164 (1.37)	18.47 (0.82)
New Fort.	New Fort., ALC XP	7 8	927 (1.72)	9.46 (1.60)
ALC	ALC, ALC XP	7 8	641 (2.48)	6.54 (2.31)

Table 1: Run time comparison.

This table shows a comparison of the run times of CLASSY using different combinations of routines. The run was 10 iterations on 4 channels of the simulated data, without any map generation. Initialization and randomization time has been subtracted off. Improvement is the speed ratio relative to the corresponding original version. (ALC means Assembly Language Coding, Fort. means Fortram.) All runs were done on a simulated data set using a 4900 point data set. The scatter in the results is due to the effects of path differences (see heading on these).

Table 2: Routine names for matrix arithmetic.

Routine	Old Fortran	New Fortran	Assembly
DOTSQ	DOTSQB	DOTSQ	DOTSQA
CORECT	CORECTB	CORECT	CORECTA
MPVS	MPVSB	MPVS	MPVSA
VPV	VPV	VPV	VPVA
VMTV	VMTV	VMTV	VMTVA
EXEC to compile	VERSOLD	VERSNEW	VERSALC

New Exponential routine (XP)

In order to speed up executions, especially when the number of channels is few, Elogic has written a special exponential routine, called XP, specialized to the needs of the CLASSY algorithm. In CLASSY, the exponential is calculated as often as is the primary quadratic form, and the standard exponential routine requires several multiplies and a divide. The new exponential used for CLASSY reduces these times by table look up and by allowing a small increase in the RMS error. The new routine has an RMS precision of about 1 part in 24,000, which is easily adequate for the needs of CLASSY. In addition, the new routine uses linear interpolation and requires only one multiply.

The new exponential routine calculates XP (x) = exp (-x/2) by subdividing the interval between 1 and 16 into 240 equal parts. A table contains the coefficients of the least-squares fit of a linear form (First - order (polynomial) to the desired exponential in this region. These coefficients give a result accurate to about one part in 24,000 RMS in the result. Arguments outside the range of 1 to 16 are handled by directly obtaining the exponential of the remaining part and multiplying it times the part calculated above. Note that because of limitations on the exponent range of the result to the routine, arguments are limited to lie between 2 log 10^{±76}. This routine will not raise an exponent underflow condition if the result is too small; rather, it returns

a result of zero. This saves additional time since STATIS will not be required to check the argument range before calculating the exponential.

The table used XP is currently calculated at execution time by the routine XPREP which must be called by CLINIT during initialization. (A version of CLINIT making this call must be used whenever the XP routine is to be used.) XP could be modified to contain these constants directly. A modified version must be also used of STATIS, which calls the XP routine.

The new exponential routine $XP(x) = \exp(-x/2)$ is thus a fast assembly language routine designed to calculate the oft-repeated exponential function quickly and with an accuracy easily sufficient for use in CLASSY.

The XP routine causes slight variations in the calculated probabilities in CLASSY, which can cause the algorithm to take a different path (See the note "Path Differences in CLASSY" below.)

XP routine

The exponential $e^{-x/2}$ is calculated as follows: x is divided into three parts, I, n, and Σ .

$$x = 16I + \frac{n}{16} + \Sigma$$
 $-22 \le I \le 21$; $0 \le n < 256$; $0 \le \Sigma < \frac{1}{16}$

Then

$$e^{-x/2}$$
 \approx XPBIG(I)*XPAR(n)*(XPD + Σ)
 \approx 0 I < -22 or I > 21

This approximation has a relative RMS error of about 1/27000, which is well adequate for the CLASSY statistical system. The variable OUTCNT counts the number of overflows made.

XPP, XPAR, and XPBIG are generated by the routine XPREP.

Equations for constants used in exponential routine:

XPAR(n) =
$$d_1e^{-2\alpha n - \alpha}$$
 n = 0(1)256
XPD = $d_0/d_1-2\alpha$
 d_0 = $\sinh \alpha/\alpha$ $\approx 1 + \frac{\alpha^2}{6} + \cdots$
 d_1 = $\frac{-3}{2\alpha^3}$ ($\sinh \alpha - \alpha \cosh \alpha$) $\approx -\frac{1}{2} - \frac{\alpha^2}{20} - \cdots$
 α = $\frac{\beta}{4} = \frac{1}{64}$ for intervals of $\frac{1}{16}$ in original argument (of $e^{-x/2}$)

These coefficients are the result of a least-squares fit of $(d_0 + d_1 \Sigma) \text{ to } e^{-\Sigma/2} \text{ for } \frac{-\beta}{2} < \Sigma < \frac{\beta}{2}$ $\text{XPBIG(I)} = e^{-8I} \qquad -23 < I \leq 22$

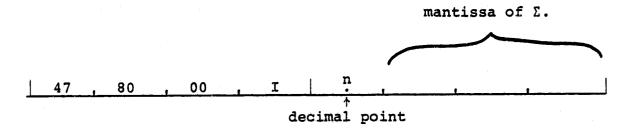
These are arranged in common block XPCOM (which is part of the exponential routine) as follows:

Quantity	Type	Bytes
OUTCNT	I * 4	4
XPXTRA	I * 4	4
XPD	R * 8	8
XPAR	R * 4	1024
XPBIG	R * 4	354

XP procedure:

The argument mantissa is shifted into the lower half of a double word (CARVE), by adding a special constant. If the argument x is $0 \le x < 16$, the upper word is fixed, and is checked. The upper byte (n) of the manification is used as an index to XPAR, and is then replaced with a standard sign-exponent byte, to yield ξ . (The non-normalized character of ξ does not matter because it is immediately added to XPD, which has a larger exponent. This also eliminates the random 4 low-order bits of ξ .) The calculation is then direct. For x outside (0,16), the procedure is the same, except for a range check and an additional factor of XPBIG, derived from an index calculated from the upper half of the double word. If x is out of range (± 352 , corresponding about to $10^{\pm 76}$), the result is set to 0, and overflows are counted.

CARVE structure:



Path Differences in CLASSY

The CLASSY algorithm may take widely different paths to the maximum likelihood solution, depending on very small effects. In some cases, whenever the clusters have not been fully resolved statistically due either to too few iterations or too little data, the two paths may end at slightly different points. Variations causing the algorithm to go on different paths include rounding error (for example use the associative and/or distributive laws) use of new standard function routines (such as the exponential routine), or other minor changes.

The reason CLASSY is so susceptible to small changes in the current version is the use of Monte-Carlo methods in processing the points with respect to classes which give them low probability. Small fluctuations in the calculations may accumulate until the threshold for the Monte-Carlo process is exceeded in one version and not in another. Since the random number in generator is sequential, this offsets the usage of all the succeeding random numbers, giving completely different Monte-Carlo choice for each point.

This, in turn, leads to moderately large fluctuations in the numerical values used, which ultimately changes the point at which the SPLIT or other threshold is passed. Since one cluster being SPLIT and separated can make a major difference in the processing of other clusters, the path taken by CLASSY rapidly becomes quite different.

In summary, minor changes in CLASSY can make large differences in the path the algorithm takes. This is mediated by the amplification caused by various thresholds, in particular those used in the Monte-Carlo subsystem. All the paths should, however, converge to the same point, except when insufficient iterations or data are used.

Appendix 1: Program listings for vector routines

This appendix contains listings of all the routines listed in Table 2. They are in the order: old Fortran, new Fortran, and assembly language. Preceding the routine listings is a listing of the EXEC routines for converting one set to the other.

SUBROUTINE CORECT (REL,PV,P,S)	CDR00010
COHHON /HISC/ HG,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT,	CDR00020
1 AMO,ODCON,XOVFLO,XUNFLO,WADJIN,ELINTH,SEPTH,VFAC,AMM,SBLTH,	C0R00030
2 INDXVL, WFAC, NPTSO, PQRATH, SPHVTH, DWFAC, GRACTH, ANDFAC,	COR00040
3 ANOHIN, AHONAX, AHORAT, VOLLIH, BIAS, PJOIN, VRJOIN, USIN, UDELSH,	C0R00050
4 BETTER, HODE, CORLEN, SPOOR	CDR00040
REAL REL(30), PV(30), S(30)	C0R00070
DO 10 I = 1, NQ	CDR00080
REL(I) = PV(I) - S(I) / P	COR00090
II = I	COR00100
C WRITE (6,9999) II,REL(I),PV(I),S(I),P	COR00110
9999 FORHAT ('CORECT I, REL, PV, S, P', 14, 4(E10.4, 2X))	CORO0120
10 CONTINUE	CORO0130
RETURN	CDR00140
END	CORO0150

```
SUBROUTINE CORECT (REL,PV,P,S)
                                                                               COR00010
      COMMON /MISC/ MQ, MM, LR, LV, MINCLS, MXAR, WTINIT, KROOT, EPS, DELT,
                                                                               C0R00020
          AND, ODCON, XOVFLO, XUNFLO, WADJIN, ELIHTH, SEPTH, VFAC, ANN, SBLTH,
                                                                               CDR00030
          INDXVL, WFAC, NPTSD, PGRATH, SPHVTH, DWFAC, GRACTH, ANDFAC,
                                                                               CDR00040
           ANONIN, ANONAX, ANORAT, VOLLIN, BIAS, PJOIN, VRJOIN, WSIH, WDELSM,
                                                                               COR00050
          BETTER, NODE, CORLEN, SPCOR
                                                                               CDR00960
      REAL REL(30), PV(30), S(30)
                                                                               CDR00070
C
                                                                               COR00080
      POV=1./P
                                                                               COR00090
      DO 10 I = 1, NQ
                                                                               C0R00100
         REL(I) = PV(I) - S(I) * POV
                                                                               COR00110
      WRITE (6,9999) I, REL(I), PV(I), S(I), P
                                                                               COR00120
9999 FORMAT ('CORECT I, REL, PV, S, P', I4, 4(E10.4, 2X))
                                                                               COR00130
10
       CONTINUE
                                                                               CDR00140
      RETURN
                                                                               CDR00150
      END
                                                                               COR00160
```

CORECTA

⇒ CORR	ESPOND	NE WAS WRITTEN BY E ING FORTRAN ROUTINE N CONTACT ELOGIC,IN	LOGIC, INC., NOV, 1979, TO REPLACE THE IN THE CLASSY SYSTEM. FOR ADDITIONAL	COR00010 COR00020 COR00030
:8		,,	,••	CUR00040
* SUBRC	TINE :	CORECT(REL_PU_P.S):	REL(I)=PV(I)-S(I)/P	COR00050
CORECT	CSECT			COR00060
0011201		CORECT, 15		COR00070
	STH	2,9,SAVE		COR00080
	LH	1,4,0(1)	ARGS ADDR	COR00090
	LA	8,4(0,0)	INCREMENT IS 1	COR00100
	EXTRN			COR00110
	L	7,=A(HISC)		COR00120
	_	NISCD,7		COR00130
	L	9,XQ	COMPARAND IS NO	COR00140
	SLA		NULTIPLY BY 4	COR00150
	SR	9,8	SUBTRACT 4	COR00160
*		.,,		COR00170
	LE	6,=21.01		COR00180
	DE	6,0(3)	GET 1/P	COR00190
*		•		C0R00200
	SR	5,5	DO LOGP INDEX I	COR00210
IC	LE	2,0(4,5)	S(I)	COR00220
	HER	2,6	MULTIPLY BY 1/P	COR00230
	LE	0,0(2,5)	PV(I)	COR00240
•	SER	0,2	SUBTRACT REG2	COR00250
	STE	0,0(1,5)	STORE REL(I)	COR00260
	BXLE	5,8,IC		COR00270
*				COR00280
	LH	2,9,SAVE		COR00290
	BR	14		CGR00300
SAVE	DS	9F		COR00310
* CONNON	/HIS	C/		COR00320
MISCD	DSECT			COR00330
NO	DS	F		CGR00340
	END			COR00350

ORIGINAL PAGE IS OF POOR QUALITY

```
FUNCTION DOTSQ(V.AMET)
                                                                                   DOT00010
         CALCULATES THE INNER PRODUCT V.V RELATIVE TO THE METRIC AMET
                                                                                   DOT00020
                                                                                   DOT00030
     DIMENSION LIST(38), XTHP(8), YTHP(7), WADJ(24), 1 INDEX(37), LSUBS(40), LSUPER(39), IDADJ(23), NSYMB(36),
                                                                            U(26),DOT00040
                                                                           OW(25),DOT00050
     2 PCUN(33), PRIRCH(32), CIN(22), CTOT(21), PROP(29), SPFAC(15), DOTO0060
3 OPROP(28), VOLIN(18), VOLRT(17), DCON(16), PORAT(14), OBEN(19), DOTO0070
        PCUH(33), PRIRCH(32),
        DISS(35), PPASS(34), PST(30), QCIN(20), PCOND(31), QPRIOR(27), DOTO0080
     5 PAVE(13), PILE(12)
      DIHENSION VRIN(475), GEN(999), GREF(999), ALINK(99)
                                                                                   DOTOO100
      EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7))
                                                                                   DOT00110
      EQUIVALENCE ( LINK(41), LSUBS(40), LSUPER(39), LIST(38),
                                                                                   DOT00120
         INDEX(37), NSYMB(36), DISS(35), PPASS(34), PCUM(33), PRIRCH(32), PCOND(31), PST(30), PROP(29), OPROP(28),
                                                                                   DOT00130
                                                                                   DOTOO140
         OPRIOR(27),
                           W(26),
                                       OW(25), WADJ(24), IDADJ(23),
                                                                                   DOT00150
          CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), VOLRT(17), DCON(16), SPFAC(15), PQRAT(14), PAVE(13),
            CIN(22),
                                                                                   DOT00160
                                                                                   DOT00170
           PILE(12), XTMP(8), YTMP(7))
                                                                                   DOT00180
      COMMON/CLUS/ JUNK(12), NARL, NTOP, NTBSZH, NWANT, LINK(14000)
                                                                                   DOT00190
      DIHENSION MXAR(31), LR(3), LV(3)
                                                                                   DOT00200
       EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT).
                                                                                   D0T00210
          (LR(3),LOVAR),(LV(1),LSUM),(LV(3),LSKEW),(LV(3),LOSUM)
                                                                                   D0T00220
C
                                                                                   DOT00230
       CONMON /MISC/ MQ, MM, LR, LV, NINCLS, MXAR, WTINIT, KROOT, EPS, DELT,
                                                                                   DOT00240
          ANQ, ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, ANN, SBLTH,
                                                                                   DOT00250
          INDXVL, WFAC, NPTSO, PQRATH, SPHVTH, DWFAC, GRACTH, ANDFAC,
                                                                                   DOT00240
           ANONIN, ANONAX, ANORAT, VOLLIN, BIAS, PJOIN, VRJOIN, USIN, UDELSH,
                                                                                   BOT00270
          BETTER, HODE
                                                                                   DOTO0280
        ( END OF STANDARD BLOCK. )
C
                                                                                   DOT00290
C
                                                                                   DOT00300
      DIMENSION NTB(32)
                                                                                   DOT00310
C
                                                                                   DOTO0320
      COHHON /STPAR/WAIT, CONLY, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI,
                                                                                   D0T00330
           PACCEL(2).NACCEL(2).VACCEL(2)
                                                                                   BOT00340
      REAL V(30), AMET(475)
                                                                                   DOT00350
       REAL+8 DDOTSQ, DGDOT
                                                                                   DOT00360
      DDOTSQ=0.
                                                                                   DOT00370
       DGDOT=V(1)+V(1)+AHET(1)
                                                                                   D0T00380
      DO 10 I=2,NQ
                                                                                   DOT00390
      MX=MXAR(I)
                                                                                   DOT00400
    7 DO 8 J=2.I
                                                                                   DOT00410
    8 DDOTSQ=DDOTSQ+V(I)+V(J-1)+AMET(MX+J-1)
                                                                                   D0T00420
   10 DGDOT=DGDOT+V(I)+V(I)+AMET(MX+I)
                                                                                   D0T00430
         THE DIAGONALS ARE HANDLED SEPARATELY BECAUSE EACH OFF-
                                                                                   BOTO0440
         DIAGONAL APPEARS TWICE, AND SO HUST BE DOUBLED.
                                                                                   DOT00450
      DDOTSQ=DDOTSQ+DDOTSQ+DGDOT
                                                                                   DOT00460
       DOTSQ = DDOTSQ
                                                                                   DOT00470
      RETURN
                                                                                   D0T00480
                                                                                   DOT00490
       END
```

```
FUNCTION DOTSO(V, ANET)
                                                                           DOT00010
        CALCULATES THE INNER PRODUCT V.V RELATIVE TO THE HETRIC ANET
                                                                           D0T00020
                                                                           D0T00030
                   LIST(38), XTHP(8), YTHP(7), WADJ(24),
                                                                     W(26),DOT90940
     1 INDEX(37), LSUBS(40), LSUPER(39), IDADJ(23), NSYHB(36),
                                                                    OW(25),DOT00050
     2 PCUM(33), PRIRCM(32),
                               CIN(22), CTOT(21), PROP(29), SPFAC(15),DOT00060
     3 OPROP(28), VOLIN(18), VOLRT(17), DCON(36), PORAT(14), ODEN(19),DOTO0070
                                PST(30), OCIN(20), PCOND(31), OPRIOR(27), DOTO0080
        DISS(35), PPASS(34),
     D0T00090
      DIHENSION VRIN(475), GEN(999), GREF(999), ALINK(99)
                                                                           D0T00100
      EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7))
                                                                           D0T00110
      EQUIVALENCE ( LINK(41), LSUBS(40), LSUPER(39), LIST(38),
                                                                           D0T00120
        INDEX(37), NSYMB(36), DISS(35), PPASS(34), PCUM(33),
                                                                           DOT00130
        PRIRCH(32), PCOND(31),
                                 PST(30), PROP(29), OPROP(28),
                                                                           D0T00140
        OPRIOR(27),
                         W(26),
                                   OW(25), WADJ(24), IDADJ(23),
                                                                           DOT00150
                     CTOT(21), OCIN(20), ODEN(19), VOLIN(18),
           CIN(22).
                                                                           DOT00160
         VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), PILE(12), XTHP( 8), YTHP( 7))
                                                                           DOT00170
                                                                           DOT00180
      COMMON/CLUS/ JUNK(12), NARL, NTCP, NTBSZH, NWANT, LINK(14000)
                                                                           D0T00190
      DIMENSION HXAR(31), LR(3), LV(3)
                                                                           D0100200
                                                                           D0T00210
      EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT),
         (LR(3),LOVAR),(LV(1),LSUN),(LV(2),LSKEW),(LV(3),LOSUN)
                                                                           DOT00220
C
                                                                           DOT00230
      COMMON /MISC/ MQ.MM.LR.LV.MINCLS.MXAR.WTINIT.KROOT.EPS.DELT.
                                                                           DOTO0240
         ANG, ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, ANN, SBLTH,
                                                                           DOT00250
         INDXVL, WFAC, NPTSO, PORATH, SPHVTH, DWFAC, GRACTH, ANOFAC,
                                                                           DOT00240
          AMOHIN, AMOHAX, AMORAT, VOLLIM, BIAS, PJOIN, VRJOIN, USIH, UDELSM,
                                                                           D0T00270
         BETTER, HODE
                                                                           D0T00280
       ( END OF STANDARD BLOCK. )
                                                                           D0T00290
                                                                           DOTO0300
      DIMENSION NTB(32)
                                                                           DOT00310
C
                                                                           DOT00320
      COMMON /STPAR/WAIT.CONLV.SKBND.SKCHI.TRBND.TRCHI.URKBND.URKCHI.
                                                                           DOT00330
                                                                           D0T00340
          PACCEL(2), MACCEL(2), VACCEL(2)
      REAL V(30), AMET(475)
                                                                           DOTO0350
      DOTSQ=V(1)+V(1)+ANET(1)
                                                                           DOT00360
      DO 10 I=2,NQ
                                                                           D0T00370
                                                                           DOT00380
      SQ=0.
                                                                           DOT00390
      MX≃MXAR(I)
    7 DO 8 J=2,I
                                                                           DOT00400
    B SQ=SQ+V(J-1) *AHET(HX+J-1)
                                                                           DDT00410
   10 DOTSQ=DOTSQ+V(I) *((V(I) *AMET(NX+I)+SQ)+SQ)
                                                                           D0T00420
        THE DIAGONALS ARE HANDLED SEPARATELY BECAUSE EACH OFF-
                                                                           D0T00430
        DIAGONAL APPEARS TWICE, AND SO MUST BE BOUBLED.
                                                                           D0T00440
      RETURN
                                                                          DOT00450
      END
                                                                           D0T00460
```

DOTSQA

* COR	RESPOND	ING FORTRAN ROUTI)	ELOGIC, INC., NOV, 1979, TO REPLACE THE NE IN THE CLASSY SYSTEM. FOR ADDITIONAL	DOTO0010 DOT00020 DOT00030
** TEL	OKUMITO	N CONTACT ELOGIC,	inc.	DOTO0040
•	AN DATE	O/N AMETA PALCINA	ATE THE INNER PRODUCT	DOTO0050
Prontil	GBLC	ara se	ADD - SUBTRACT PRECISION	D0100060
3P	SETC	'ER'	ER SINGLE PRECISION; DR- DOUBLE PREC	
DOTSO	CSECT		EN STROEF LUFOTSTOR'S DIV. DOODER LUFD	D0100080
DUISU		DOTSQ,15		DOT00090
	STM	2,12,SAVE		DOTO0100
	LH	1,2,0(1)	ARGS ADDR	D0T00110
	LA	4,4(0,0)	I INCREMENT IS !	DOT00120
	EXTRN		e minimient wit :	D0T00130
	L	11,=A(HISC)		DBT00140
		HISCD, 11		DOT00150
	Ĺ	5, MQ	I COMPARAND IS NO	DOT00160
	SLA	5,2	HULTIPLY BY 4	D0T00170
:#			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	D0T00180
	LR	6.4	J INCREMENT IS 1	D0T00190
	LA	8,8(0,0)		D0T00200
	SR	1,8	INDEX OF V STARTS FROM 1	D0T00210
	SR	2,8	INDEX OF AMET STARTS FROM 1	D0T00220
:#		,	•	D0T00230
	LE	0,4(1,4)	V(1)	D0T00240
	HER	0,0	V(1)*V(1)	D0T00250
	HE	0,4(2,4)	*AHET(1)	D0T00260
	S&P	2,2	0.	DOT00270
塘		·		D0T00280
	LR	7,8	DO LOOP INDEX I STARTS FROM 2	D0T00290
I1	LAP	4,2	SQ=0.	D0T00300
	L	12, MXAR-4(7)	HX=HXAR(I)	D0T00310
	SLA	12,2	HULTIPLY BY 4	DOT00320.
	AR	12,2	ADD ADDR AMET	D0T00330
:#				D0T00340
	LR	9,8	DO LOOP INDEX J STARTS FROM 2	D0T00350
12	LE	6,0(1,9)	V(J-1)	D0T00360
	ME	6,0(12,9)	V(J-1)*AHET(HX+J-1)	D0T00370
	A&P	4,6	SQ	D0T00380
	BXLE	9,6,12		DOT00390
				DDT00400

: 4				
	LE	6,4(1,7)	V(I)	D0T00410
	HE	6,4(12,7)	V(I) + AMET(MX+I)	D0T00420
	ALP	6,4	+SQ	00100430
	ALP	6,4	+50	DOT00440
	HE	6,4(1,7)	*V(I)	D0T00450
	ALP	0,6	DOTSQ	DOT00460
	BXLE	7,4,11		DOT00470
ife				DOT00480
	LH	2,12,5AVE		DOT00490
	BR	14		DOT00500
*				BOT00510
SAVE	DS	12F		DOT00520
; #				D0T00530
\$ \$				DOT00540
* CONMON	/HIS	C/		D0T00550
MISCD	DSECT			DOT00540
MQ	DS	F		00700570
MH	DS	F		DOT00580
LR	DS	3F		D0T00590
LV	DS	3F		0000000
NINCLS	DS	F		DOT00610
MXAR	DS	31F		D0T00620
	END			D0T00630

```
SUBROUTINE HPVS(AM,C,V)
                                                                               MPV00010
C
          SETS AH=AH+V+V+C
                               (TENSOR PRODUCT)
                                                                               HPV00020
C
                                                                               HPV00030
      COMMON /MISC/ MQ, MM, LR, LV, NINCLS, MXAR, WTINIT, KROOT, EPS, DELT,
                                                                               HPV00040
         AND, ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, AMM, SBLTH,
                                                                               HPV00050
         INDXVL, UFAC, NPTSO, PGRATH, SPHVTH, DUFAC, GRACTH, AMOFAC,
                                                                               MPV00060
          ANOHIN, AHOHAX, AHORAT, VOLLIH, BIAS, PJOIN, VRJOIN, WSIH, WDELSH,
     3
                                                                               HPV00070
          BETTER, MODE, CORLEN, SPCOR
                                                                               MPV000B0
C
                                                                               HPV00090
      REAL AM(475), V(30)
                                                                              MPV00100
      LOC=0
                                                                              HPV00110
      DO 10 I=1,NQ
                                                                              HPV00120
      DO 10 J=1,I
                                                                              HPV00130
      LOC=LOC+1
  10 AH(LOC)=AH(LOC)+V(I)+V(J)+C
                                                                              HPV00140
                                                                              HPV00150
      RETURN
                                                                              MPV00160
      END
                                                                              HPV00170
```

```
SUBROUTINE MPVS(AM,C,V)
                                                                              MPV00010
C
         SETS AN=AN+V*V*G (TENSOR PRODUCT)
                                                                              MPV00020
C
                                                                              HPV00030
      CONNON /MISC/ MQ, MM, LR, LV, MINCLS, MXAR, WTINIT, KROOT, EPS, DELT,
                                                                              MPV00040
         ANO, ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, ANN, SBLTH,
                                                                              NPV00050
         INDXVL, WFAC, NPTSO, PORATH, SPHVTH, DWFAC, GRACTH, AMOFAC,
                                                                              NPV00040
           ANONIN, ANONAX, AMORAT, VOLLIH, BIAS, PJOIN, VRJOIN, USIM, UDELSM.
                                                                              MPV00070
         BETTER, NODE
                                                                              NPV00080
Č
                                                                              MPV00090
      REAL AM(475), V(30)
                                                                              HPV00100
      LOC=0
                                                                              NPV00110
      DO 10 I=1,MQ
                                                                              HPV00120
      CV=C+V(I)
                                                                              MPV00130
      DO 10 J=1,I
                                                                              MPV00140
      LOC=LOC+1
                                                                              NPV00150
   10 AM(LOC)=AM(LOC)+V(J)+CV
                                                                              HPV00160
      RETURN
                                                                              MPV00170
      END
                                                                              MPV00180
```

HPVSA

THIS ROUTINE WAS URITTEN BY ELOGIC, INC., NOV, 1979, TO REPLACE THE CORRESPONDING FORTRAN ROUTINE IN THE CLASSY SYSTEM. FOR ADDITIONAL NPV00020 INFORMATION CONTACT ELOGIC, INC. SUBROUTINE HPVS(AM,C,V): SETS AM=AM+V+V+C MPV00050
INFORMATION CONTACT ELOGIC, INC. # SUBROUTINE HPVS(AH,C,V): SETS AM=AM+V+V+C HPVS CSECT USING HPVS,15 STH 2,12,SAVE LH 1,3,0(1) ARGS ADDR LA 4,4(0,0) I INCREHENT IS 1 MPV00030 MPV00030 MPV00050 MPV00050 MPV00090 MPV00090 MPV00090
SUBROUTINE HPVS(AN,C,V): SETS AN=AN+V*V*C HPV00050 HPVS CSECT HPV00060 USING HPVS,15 HPV00070 STH 2,12,SAVE LN 1,3,0(1) ARGS ADDR LA 4,4(0,0) I INCREHENT IS 1 HPV00100
SUBROUTINE HPVS(AN,C,V): SETS AM=AM+V*V*C MPV00050 HPVS CSECT MPV00060 USING HPVS,15 MPV00070 STH 2,12,SAVE LN 1,3,0(1) ARGS ADDR LA 4,4(0,0) I INCREHENT IS 1 MPV00100
MPVS CSECT MPV00060 USING MPVS,15 MPV00070 STH 2,12,SAVE MPV00080 LN 1,3,0(1) ARGS ABUR MPV00090 LA 4,4(0,0) I INCREMENT IS 1 MPV00100
USING HPVS,15 MPV00070 STM 2,12,SAVE HPV00080 LN 1,3,0(1) ARGS ADDR HPV00090 LA 4,4(0,0) I INCREHENT IS 1 MPV00100
STH 2,12,5AVE HPV00080 LN 1,3,0(1) ARGS ADDR HPV00090 LA 4,4(0,0) I INCREHENT IS 1 HPV00100
LH 1,3,0(1) ARGS ADDR HPV00090 LA 4,4(0,0) I INCREHENT IS 1 HPV00100
LA 4,4(0,0) I INCREMENT IS 1 MPV00100
EXIKM MISC HEVOLIN
L 12.=à(HISC) MPV00120
- · · · · · · · · · · · · · · · · · · ·
L 5, HQ COMPARAND IS HQ MPV00140
SLA 5,2 MULTIPLY BY 4 MPV00150
SR 5,4 SUBTRACT 4 MPV00160
** NPV00170
LR 8,4 J INCREMENT IS 1 MPV00180
SR 7,7 LOC HPV00190
** NPV00200
SR 9,9 DO LOOP INDEX I MPV00210
LE 2,0(2) C MPV00220
MPV00230
II LER 6,2 C MPV00240
HE 6,0(3,9) C+V(I) NPV00250
HPV00260
SR 11,11 DO LOOP INDEX J HPV00270
12 LER 0,6 C*V(I) MPV00280
HE 0,0(3,11) C+V(I)+V(/) HPV00290
AE 0,0(1,7) ABD AM(LGC) NPV00300
STE 0,0(1,7) STORE AH(LDC) HPV00310
AR 7,4 ADD 1 TO LOC HPV00320
BXLE 11,8,12 HPV00330
BXLE 9,4,11 HPV00340
** MPV00350
LH 2,12,5AVE HPV00360
BR 14 MPV00370
SAVE DS 12F MPV00380
* CONHON /HISC/ MPV00390
MISCD DSECT MPV00400
NQ DS F NPV00410
END NPV00420

1

	SUBROUTINE VNTV(VA,AMET,VB)	VHT00010
Ċ	SETS VA=AMET+VB	VMT00020
-	COMMON /MISC/ KG, MM, LR, LV, MINCLS, MXAR, WTINIT, KROOT, EPS, DELT,	VHT00030
	1 ANG.ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, ANN, SBLTH,	VHT00040
	2 INDXVL, WFAC, NPTSO, PQRATH, SPNVTH, DWFAC, GRACTH, ANDFAC,	VMT00050
	3 AHOHIN, AHOHAX, AHORAT, VOLLIH, BIAS, PJOIN, VRJOIN, USIN,	09000HA
	4 BETTER, MODE, CORLEN, SPCOR	VHT00070
	REAL VA(30), VB(30), AHET(475)	VHTOOOBO
	LOCA=0	VNT00090
	DO 20 I=1,NQ	VHT00100
	SUN=0.	VMT00110
	DO 10 J=1.I	VHT00120
	LOCA=LOCA+1	VHT00130
	10 SUN=SUN+ AMET(LOCA) + VB(J)	VHT00140
	IF(I.EQ.NQ) GO TO 20	VHT00150
	JS=I+1	VMT00160
	LOCB=LOCA+I	VHT00170
	DO 11 J=JS,NQ	VHT00180
	SUN=SUN+ANET(LOCB)*VB(J)	VNT00190
	11 LOCB=LOCB+J	VHT00200
	20 VA(I)=SUM	VHT00210
	RETURN	VHT00220
	END	VNT00230

ORIGINAL PAGE IS

AVTHV

# THIS	ROUTI	NE WAS WRITTEN	BY ELOGIC, INC., NOV, 1979, TO REPLACE	VHT00010
			TINE IN THE CLASSY SYSTEM. FOR ADDITIONAL	VHT00020
* INFO	RMATIO	N CONTACT ELOGI	C, INC.	OE0001HV
t ķ				VHT00040
* SUBROU	TINE	UNTU(UA.AMET.UB): SETS VA=AHET+VB	VHT00050
VHTV	CSECT	,		00000THV
,,		VHTV,15		VHT00070
	STH	2,12,SAVE		08000THV
	LH	1,3,0(1)	ARGS ADDR	VHT00090
	LA	4,4(0,0)	I INCREMENT IS 1	VHT00100
	EXTRN		1 Induction 14)	VNT00110
	L	7,=A(HISC)		VMT00120
	_	HISCD.7		VMT00130
	L	5,HQ	COMPARAND IS HO	VHT00140
	SLA	5,2	HULTIPLY BY 4	VMT00150
	DROP	7	MODITED DI 4	VNT00150
:4:	DNUF	/		VMT00170
٠,٠	LR	8,4	J INCREMENT IS 1	08100170
			INDEX OF VA STARTS FROM 1	
		1,4		VNT00190
	SR	2,4	INDEX OF ANET STARTS FROM 1	
	SR	3,4	INDEX OF VB STARTS FROM 1	VMT00210
	SR	7,7	INITIALIZE LOCA	VNT00220
*				VMT00230
	LR	9,4	DO LOOP INDEX I STARTS FROM 1	VMT00240
It	LE	6,=E'0.0'	INITIALIZE SUN	VNT00250
1#				VMT00260
	LR	11,4	DO LOOP INDEX J STARTS FROM 1	VHT00270
12	AR	7,4	ADD 1 TO LOCA	VHT00280
	LE	0,0(2,7)	AMET(LOCA)	VHT00290
	ME	0,0(3,11)	HULTIPLY BY VB(J)	VNT00300
	AER	6,0	ADD TO SUM	01E00THV
	BXLE	11,8,12		VHT00320
:#				OEEOOTHV
	CR	9,5	coh. are index i with its comparand	VHT00340
	BZ	A12		VHT00350
	LR	11,9	I	09E001HA
	AR	11,4	JS=I+1	VNT00370
	LR	12,7	LOCA	VMT00380
	AR	12,9	LOCA +I	VNT00390
13	LE	0,0(2,12)	AMET(LOCB)	VHT00400
	HE	0,0(3,11)	MULTIPLY BY VB(J)	VNT00410
	AËR	6,0	ADD TO SUK	VNT00420
	AR	12,11	ADD J TO LOCB	0E400THV
	BXLE	11,4,13	,,, ,, ,,, ,,,,,,,,,,,,,,,,,,,,,,,,,,,	VHT00440
1		,.,		VNT00450
A12	STE	6,0(1,9)	VA(I)	VMT00460
****	BXLE	9,4,11	*****	VHT00470
:#				VHT00480
•	LN	2,12,SAVE		VNT00496
	BR	14		VHT00500
SAVE	DS	12F		VNT00510
* CONHON				VMT00529
MISCD	DSECT	V/		VNT00530
MO HISCD	DS	F		VNT00540
118.	END	•		VNT00550
	FWN			ALLIADADAA

page 24

	SUBROUTINE VPV(VA, FAC, VB)	VPV00010
C	SETS VA=VA+FAC+VB	VPV00020
C		VPV00030
_	COMMON /MISC/ MQ, MM, LR, LV, MINCLS, MXAR, WTINIT, KROOT, EPS, DELT,	VPV00040
	1 AMO.ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, AMM, SBLTH,	VPV00050
	2 INDXVL, WFAC, NPTSO, PORATH, SPHVTH, DWFAC, GRACTH, AMOFAC,	VPV00040
	3 AMONIN, AMONAX, AMORAT, VOLLIM, BIAS, PJOIN, VRJOIN, WSIM, WDELSM,	VPV00070
	4 BETTER, MODE, CORLEN, SPCOR	VPV00080
C	,,,,,,,	VPV00090
	REAL VA(30), VB(30)	VPV00100
	DO 10 I=1,HQ	VPV00110
	10 VA(I)=VA(I)+VB(I)+FAC	VPV00120
	RETURN	VPV00130
	END	VPV00140

* THIS	ROUTI	ne was written b	Y ELOGIC, INC., NOV, 1979, TO REPLACE THE	VPV00010
* CORR				VPV00020
* INFO	INFORMATION CONTACT ELOGIC, INC.			VPV00030
*				VPV00040
* SUBROU	SUBROUTINE VPV(VA, FAC, VB): VA(I)=VA(I)+FAC*VB(I)			VPV00050
VPV	CSECT	• • •		VPV00060
	USING	VPV,15		VPV00070
	STH	2,7,SAVE		VPV00080
	LH	1,3,0(1)	ARGS ADDR	VPV00090
	LE	2,0(2)	ARG2	VPV00100
	LA	4,4(0,0)	INCREMENT IS 1	VPV00110
	EXTRN	MISC		VPV00120
	L	7,=A(HISC)		VPV00130
	USING	MISCD,7		VPV00140
	L	5, HQ	COMPARAND IS MQ	VPV00150
	SLA	5,2	HULTIPLY BY 4	VPV00160
	SR	5,4	SUBTRACT 4	VPV00170
: #		,		VPV00180
	SR	2,2	DO LOOP INDEX I STARTS FROM 1	VPV00190
IC	LER	0,2	FAC	VPV00200
	HE	0,0(3,2)	FAC+VB(I)	VPV00210
	AE	0,0(1,2)	VA(I)	VPV00220
	STE	0,0(1,2)	STORE VA(I)	VPV00230
	BXLE	2,4,IC	BRANCH LOCATION	VPV00240
*				VPV00250
•	LH	2,7,SAVE	*	VPV00260
	BR	14		VPV00270
SAVE	DS	7F		VPV00280
* CONHON		C/		VPV00290
MISCD	DSECT			VPV00300
MQ	DS	F		VPV00310
	END			VPV00320

Appendix 2: Program listings for the fast exponential system.

This appendix contains the routines used by the fast exponential system. They include the routine XP, the table generator XPREP, the modified version of CLINIT, showing where XPREP is called; and the modified version of STATIS, showing the calls to XP. (The version of STATIS displayed also contains modifications for the additional statistics collection, which will be the subject of an additional report.) The modification to CLINIT consists only of the one line calling XPREP, which may be executed even if XP is not used. Elogic recommends that the version containing the call to XPREP be used at all times.

		N CONTACT ELOGIC, IN	IN THE CLASSY SYSTEM. FOR ADDITIONAL NC.	ΧP	00020 00030 00040
-	NTIAL	ROUTINE, EXP(-X/2)			00050
XP	CSECT				00060
•••	USING				00070
	L	1,0(1)	ARG ADDR		00080
	LE	0,0(1)	ARG		00090
	AD	O,SHIFWRD	EXTRACT MANTISSA		00100
	STD	O, CARVE	=-, · · · · · · · · · · · · · · · · · · ·		00110
	L	1,CARVE	GET UPPERWORD		00120
		1.UPCARVE	CHK ARG BETWEEN		00130
	BNZ	NSEXP	0 AND 16		00140
XPSEQ	IC	1,CARVE+4	GET UPPER BYTE	XP	00150
		CARVE+4,X'3F'	PUT IN EXPONENT	XΡ	00160
	LE	0,CARVE+4	LOAD XI		00170
	AE	O,XPD	CALC EXPRESSION	XP	00180
	ALR	1,1	INDEX	ΧP	00190
	^LR	1,1		ΧP	00200
	HE	0,XPAR(1)	EXPONENTIAL TABLE	ΧP	00210
	BR	14	(BYE)	ΧP	00220
: #:				XP	00230
NSEXP	C	1, NINCARV	CHK UNDERFLOW	ΧP	00240
	BM	ZEROIT		XP	00250
	C	1, HAXCARV	CHK OVERFLOW	ХЪ	00260
	BP	OUTRANGE		XΡ	00270
	ŚR	1,1		ΧP	00280
		1,CARVE+4	(DUP XPSEQ)	ΧP	00290
	HVI	CARVE+4,X'3F'		ΧP	00300
	LE	0,CARVE+4			00310
	AE	O,XPD			00320
	ALR	1,1			00330
	ALR	1,1			00340
	HE	0,XPAR(1)			00350
	L	1,CARVE	GET BIG PART		00360
	ALR	1,1	INDEX -USES SPEC		00370
	ALR	1,1	PROPERTIES OF SHIFURD		00380
	HE	O,XPBIG(1)	BIG EXPONENTIAL TABLE		00390
	BR	14	(BYE)		00400
:\$					00410
ZEROIT	LE	0,=F′0′	UNDERFLOW -		00420
	BR	14	RETURN O.		00430
*					00440
OUTRANGE		1,OUTCNT			00450
	A	1,=F′1′			00460
	ST	1,OUTCNT			00470
	В	ZEROIT		XĽ	00480

18				***	00490
: #				XP	00500
CARVE	DS	D		XP	00510
SHIFWRD	DC	A(X'47800000')	SHIFTS ARGUMENT TO GET	XΡ	00520
	DC	F'0'	HANTISSA	XP	00530
UPCARVE	EQU	SHIFURD	ELINS SHIFWRD	ΧP	00540
MINCARV	DC	F'-21'		XΡ	00550
MAXCARV	DC	F'21'		XP	00560
14				XP	00570
XPCON	DS	OD		XP	00580
*** ==/,	ENTRY	XPCOM		XP	00590
OUTCHT	DC	F'0'		XP	
XPXTRA	DC	F/0/		XP	
XPD	DS	D		XP	
XPAR	DS	256F		XP	
701 IIII	DS	21F	XPBIG NEG ARG	XP	00640
XPBIG	DS	22F	ni see mee ning	XP	00650
F1 010	END	6 to 1			00660

page 28

page 29

	SUBROUTINE XPREP	XPR00010
	CONHON /XPCOM/ OUTCHT,XPXTRA,XPD,XFARO(256),XPBIGO(43)	XPR00020
	REAL*8 XPD, ALPHA, C2, D1	XPR00030
C		XPR00040
	BETA=1./16.	XPR00050
	ALPHA=BETA/4.	XPR00060
	C2=2.*ALPHA	XPR00070
	D1=3.*(DSINH(ALPHA)-ALPHA*DCOSH(ALPHA))/(C2*ALPHA*ALPHA)	XPR00080
	XPD=DSINH(ALPHA)/(ALPHA*D1) - C2	XPR00090
	DO 10 N=1.256	XPR00100
10	XPARO(N)=D1*DEXP(-C2*(N-1)-ALPHA)	XPR00110
	DO 20 I=1,43	XPR00120
20	XPBIGO(I)=DEXP(-8.D0*(I-22))	XPR00130
	RETURN	XPR00140
	FND	YPR00150

C THIS ROUTINE CONTAINS THE VARIOUS STATEMENTS NECESSARY TO CL100030 CL100330 CL1003		SUBROUTINE CLINIT(KROT)	CL TAAA LA
C INITIALIZE THE CLUSTERING ALGORITHM. CL100030 DIHENSION LIST(38), XTHP(8), YTHP(7), WADJ(24), W(25), CL100030 1 INDEX(37), LSUBS(40), LSUPER(37), IDADJ(23), NSYHB(36), 0W(25), CL100060 2 POLIN(33), PRIRCH(32), CIN(22), CTOT(21), PROP(29), SPFAC(15), CL100070 3 OPROP(28), VOLIN(18), VOLRT(17), DCON(16), PQRAT(14), ODEN(17), CL100080 4 DISS(35), PPASS(34), PST(30), OCIN(20), PCOND(31), OPRIOR(27), CL100080 5 PAVE(13), PILE(12) DIHENSION WRIN(475), GEN(797), GREF(797), ALINK(79) CL1001100 EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7)) CL100120 EQUIVALENCE (LINK(41), LSUBS(40), LSUPER(37), LIST(38), CL100130 1 INDEX(37), NSYHB(36), DISS(35), PPASS(34), PCUN(33), CL100130 2 PRIRCH(32), PCOND(31), PST(30), PROP(27), OPROP(28), CL100150 3 OPRIOR(27), W(26), OW(25), WADJ(24), IDADJ(23), CL100150 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CL100150 5 VOLRT(17), DCON(16), SPFAC(15), PPRAT(14), PAVE(13), CL100160 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CL100160 6 PILE(12), XTHP(8), YTHP(7)) COHMON/CLUS/JUNK(12), NARL,NTOP,NTBSZH,NWANT,LINK(14000) CL100220 DIHENSION MXAR(31),LR(3),LV(3) CL100210 COHMON/CLUS/JUNK(12),NARL,NTOP,NTBSZH,NWANT,LINK(14000) CL100220 DIHENSION MXAR(31),LR(3),LV(3) CL100210 COHMON /MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITHIT, KROOT, EPS, DELT, CL100220 COHMON /MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITHIT, KROOT, EPS, DELT, CL100220 CC COHMON /MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITHIT, KROOT, EPS, DELT, CL100220 CC COHMON /SOFLO, XUMFLO, WADJIN, ELINTH, SEPTH, VFAC, AMM, SBLTH, CL100220 CC COHMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, WKBND, URKCHI, LO0330 CC COHMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, WKBND, URKCHI, LO0330 CC COHMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, WKBND, URKCHI, LO0330 CC COHMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, WKBND, URKCHI, LO0330 CC COHMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, WKBND, URKCHI, LO03300 CL100330 CONLORD / TRUE CONL	r	· · · · · · · · · · · · · · · · · · ·	
DIHENSION LIST(38), XTHP(8), YTMP(7), WADJ(24), W(26), CLI00050			
DIHENSION LIST(38), XTMP(8), YTMP(7), WADJ(24), U(24), CLI00050 I INDEX(37), LSUBS(40), LSUPER(37), IDADJ(23), NSYMB(36), UM(25), CLI00060 PCUM(33), PRIRCN(32), CIN(22), CTOT(21), PROP(29), SPFAC(15), CLI00060 A DISS(35), PPASS(34), PST(30), OCIN(20), PCOND(31), OPRIOR(27), CLI00090 A DISS(35), PPASS(34), PST(30), OCIN(20), PCOND(31), OPRIOR(27), CLI00090 DIHENSION VRIN(475), GEN(999), GREF(999), ALINK(99) CLI00110 EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7)) CLI001120 EQUIVALENCE (LINK(41), LSUBS(40), LSUPER(39), LIST(38), CLI00130 I INDEX(37), NSYMB(36), DISS(35), PPASS(34), PCUN(33), CLI00140 PRIRCH(32), PCOND(31), PST(30), PROP(29), OPROP(28), CLI00150 3 OPRIOR(27), U(26), OU(25), UADJ(24), IDADJ(23), CLI00150 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CLI00160 5 VOLRT(17), DCON(16), SPFAC(15), PQRAT(14), PAVE(13), CLI00160 6 PILE(12), XTHP(8), YTHP(7) CLI00190 CDHHON/CLUS/ JUNK(12), MARL, MTOP, NTBSZH, HUANT, LINK(14000) CLI00220 DIHENSION HXAR(31), LR(3), LV(3) CLI00110 CUMPAR (10), CLIV(11), LSUM), (LV(2), LSKEW), (LV(3), LOSUH) CLI00220 CLI00220 CLI00230	0	INTITALIZE INE CLUSIERIRO ALGORIIAN.	
INDEX(37), LSUBS(40),LSUPER(37), IDADJ(23), NSYBB(36), OU(25),CL100060 2 PCUM(33),PRIRCH(32), CIM(22), CTOT(21), PROP(29), SPPACIS,CL100090 4 DISS(35), PPASS(34), PST(30), OCIN(20), PCOMB(31),OPRIDR(27),CL100090 5 PAWE(13), PILE(12) DITHENSION VRIN(475),GEN(999),GREF(999),ALINK(99) CL100110 EQUIVALENCE (LINK(41),ALINK(41),GREF(3),GEN(7),VRIN(7)) CL100110 2 PRIRCH(32),PCOND(31), PST(30),PROP(29),QRDP(28), CL100150 3 OPRIGR(27), U(26), OU(25), UADJ(24),IDADJ(23), CL100150 4 CIN(22),CTOT(21),OCIN(20),ODEN(19),VOLIN(18), CL100160 4 CIN(22),CTOT(21),OCIN(20),ODEN(19),VOLIN(18), CL100160 5 VOLRT(17),DCON(16),SPFAC(15),PORAT(14),PAVE(13), CL100180 6 PILE(12),XTMP(8),YTMF(7) CL100180 6 PILE(12),XTMP(8),YTMF(7) CL100180 6 PILE(12),XTMP(8),YTMF(7) CL100190 DIHENSION HARAK(31),LR(3),LV(3) CL100190 DIHENSION HARAK(31),LR(3),LV(3) CL100190 COMHON/CLUS/ JUNK(12),NARL,NTOP,NTBSZN,HUANT,LINK(14000) CL100220 CL100210 CL100210 CL100220 CL100230 CL100210 CL100210 CL100220	L	DIMENSION LICTURES VINDI AL VINDI IN HABILARA (1707)	
2 PCUN(33), PRIRCH(32), CIN(22), CTOT(21), PROP(29), SFFAC(15), CL100070 3 OPROP(28), VOLIN(18), VOLRT(17), DCON(16), PRAT(14), DDEN(19), CL100080 4 DISS(35), PPASS(34), PST(30), OCIN(20), PCOND(31), DPRIOR(27), CL100090 5 PAVE(13), PILE(12) DIMENSION VRIN(475), GEN(999), GREF(999), ALINK(99) CL100110 EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7)) CL100120 EQUIVALENCE (LINK(41), ASUBS(40), SUPER(39), LIST(38), CL100130 1 INDEX(37), NSYHB(36), DISS(35), PPASS(34), PCUN(33), CL100140 2 PRIRCH(32), PCOND(31), PST(30), PROP(29), OPROP(28), CL100150 3 OPRIOR(27), U(26), OU(25), UADJ(24), IDADJ(23), CL100160 4 CIN(22), CTOT(21), OCIN(20), DBEN(19), VOLIN(18), CL1001160 5 VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), CL100180 6 PILE(12), XTHP(8), YTHP(7)) COMHON/CLUS7, JUNK(12), MARL, NTOP, NTBSZH, NUANT, LINK(14000) DIHENSION HXAR(31), LR(3), LV(3) EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT), CCOHHON /HISC/ HG, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CL100220 1 (LR(3), LOVAR), (LV(1), LSUH), (LV(2), LSKEW), (LV(3), LOSUH) CCOHHON /HISC/ HG, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CL100240 CL100270 CL100270 CL100270 CL100270 CL100270 CL100310 CL100330 CL100340		UINENSIUM LISI(38), XIMP(8), YIMP(7), WAUJ(24), W(26)	,01100050
3 OPROP(28), VOLIN(18), VOLRT(17), DCON(16), PORAT(14), ODEM(19), CLI00090 4 DISS(35), PPASS(34), PST(30), OCIN(20), PCOND(31), OPRIOR(27), CLI00090 5 PAVE(13), PILE(12) CLI00110 DIMENSION VRIN(475), GEN(999), GREF(999), ALINK(99) CULI00110 EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7)) CLI001130 1 INDEX(37), NSYHB(36), DISS(35), PPASS(34), PCUN(33), CLI00130 2 PRIRCH(32), PCOND(31), PST(30), RROP(29), OPROP(28), CLI00150 3 OPRIOR(27), U(26), OU(25), UADJ(24), IDADJ(23), CLI00160 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CLI00170 5 VOLRT(17), DCON(16), SPFAC(15), PRAT(14), PAVE(13), CLI00170 6 PILE(12), XTHP(8), YTHP(7) COMHON/CLUS/ JUNK(12), NARL, NTOP, NTBSZH, NUANT, LINK(14000) CLI00200 DIHENSION MXAR(31), LR(3), LV(3) EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT), CLI00220 CLI00210 COMHON /HISC/ HB, HH, LR, LV, NINCLS, HXAR, WITINIT, KROOT, EPS, DELT, CLI00250 CLI00230 CLI002		1 INVEX(37), LSUBS(40), LSUPER(39), 19AUJ(23), NSYMB(36), UW(25)	,CL100060
A DISS(35), PPASS(34), PST(30), OCIN(20), PCOND(31), OPRIOR(27), CL100100 DIMENSION VRIN(475), GEN(999), GREF(999), ALINK(99) CL1001100 EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7)) CL100120 EQUIVALENCE (LINK(41), LSUBS(40), LSUPER(39), LIST(38), CL100130 INDEX(37), NSYHS(36), DISS(35), PPASS(34), PCUN(33), CL100140 2 PRIRCK(32), PCOND(31), PST(30), PROP(29), OPROP(28), CL100150 3 OPRIOR(27), W(26), OW(25), WADJ(24), IDADJ(23), CL100160 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CL1001/0 5 VOLRT(17), DCONK(16), SPFAC(15), PQRAT(14), PAVE(13), CL100180 6 PILE(12), XTMP(8), YTMP(7), CL100180 6 PILE(12), XTMP(8), YTMP(7), CL100180 CUMHON/CLUS/ JUNK(12), NARL, NTOP, NTBSZH, NWANT, LINK(14000) CL100200 DIMENSION HXAR(31), LR(3), LV(3) CL100210 EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT), CL100220 CL100210 CL100220		2 PCUM(33), PRIRCH(32), CIM(22), CTOT(21), PROP(29), SPFAC(15)	,CL100070
S PAVE(13), PILE(12)			
DIMENSION VRIN(475),GEN(999),GREF(999),ALINK(99) EQUIVALENCE (LINK(41),ALINK(41),REF(8), GEN(7),VRIN(7)) CL100120 EQUIVALENCE (LINK(41),LSUBS(40),LSUPER(39), LIST(38), CL100130 1 INDEX(37), NSYHB(36), DISS(35), PPASS(34), PCUN(33), CL100140 2 PRIRCH(32), PCOMB(31), PST(30), PROP(29), OPROP(28), CL100150 3 OPRIOR(27), U(28), OU(25), UADJ(24), IDADJ(23), CL100160 4 CIN(22), CTOT(21), OCIM(20), ODEN(19), VQLIN(18), CL1001/0 5 VOLRT(17), DCOM(16), SPFAC(15), PORAT(14), PAVE(13), CL100180 6 PILE(12), XTHP(8), YTHP(7)) COMHOM/CLUS/ JUNK(12),NARL,NTOP,NTBSZH,NWANT,LINK(14000) CL100220 DIHENSION HXAR(31),LR(3),LV(3) EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), CL100220 1 (LR(3),LOUAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) CL100230 COMHON /HISC/ HQ,HM,LR,LV,NINCLS,HXAR,WITINIT,KROOT,EPS,DELT, CL100250 1 AMQ,DDCON,XOVFLO,XUNFLO,WADJIN,ELINTH,SEPTH,WFAC,ANM,SBLTH, CL100250 1 ANDAUL,MFAC,NPTSO,PQRATH,SPHVTH,DUFAC,GRACTH,AHOFAC, CL100270 3 ANOMIN,AHOMAX,AHORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIN,WDELSH, CL100280 C (END OF STANDARD BLOCK.) C DUBBLE PRECISION XTEMP,YTEMP,ZTEMP,DURK,DURKD C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, 1 PACCEL(2),MACCEL(2),VACCEL(2) C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, 1 LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, CL100330 C COHHON/CLUSTR/ IBEGIN,TOTURD,CLSNAH,IPT,NOFLD, SYH(61), CL100330 C COHHON/STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CL100330 C COHHON/STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CL100330 C COHHON/STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CL100330 C C C C C C C C C C C C C C C C C C C			
EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7)) EQUIVALENCE (LINK(41), LSUBS(40), LSUPER(39), LIST(38), CLI00130 1 INDEX(37), NSYHB(36), DISS(35), PPASS(34), PCUN(33), CLI00140 2 PRIRCH(32), PCOND(31), PST(30), PROP(29), OPROP(28), CLI00150 3 OPRIDR(27), U(26), OU(25), UADJ(24), IDADJ(23), CLI00160 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CLI001/0 5 VOLRT(17), DCON(16), SPFAC(15), PORAT(14), PAVE(13), CLI00180 6 PILE(12), XTMP(8), YTMP(7)) COHHON/CLUS/ JUNK(12), NARL, NTOP, NTBSZH, NWANT, LINK(14000) CLI00210 DIMENSION MXAR(31), LR(3), LU(3) CLI00210 EQUIVALENCE (LR(1), LURIN), (LR(2), LKURT), CLI00220 1 (LR(3), LOVAR), (LV(1), LSUH), (LV(2), LSKEW), (LV(3), LOSUH) CLI00220 COHHON / MISC/ MG, HM, LR, LV, NINCLS, HXAR, WIINIT, KROOT, EPS, DELT, CLI002240 COHHON / MISC/ MG, HM, LR, LV, NINCLS, HXAR, WIINIT, KROOT, EPS, DELT, CLI002260 CINO2440 CINO2440 CINO2450 CINO2450 CINO2450 CINO2450 CINO2450 CINO2450 CINO2450 CINO2450 CINO2450 CLI00250 CLI0			
EQUIVALENCE (LINK(41), LSUBS(40), LSUPĒR(39), LIST(38), CLI00130 1 INDEX(37), NSYHB(36), DISS(35), PPASS(34), PCUR(33), CLI00140 2 PRIRCH(32), PCOND(31), PST(30), PROP(29), DROP(29), CLI00150 3 OPRIDR(27), U(26), OU(25), UADJ(24), IDADJ(23), CLI00160 4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18), CLI00160 5 VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), CLI00180 6 PILE(12), XTHP(8), YTHP(7) CLI00190 COHHON/CLUS/ JUNK(12), NARL, NTOP, NTBSZH, NWANT, LINK(14000) CLI00200 DIHNBISTON HXAR(31), LR(3), LV(3) CLI00210 EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT), CLI00220 1 (LR(3), LOVAR), (LV(1), LSUH), (LV(2), LSKEW), (LV(3), LOSUH) CLI00230 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00240 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00240 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00230 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00240 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00230 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00230 C COHHON / MISC/ HQ, HH, LR, LV, NINCLS, HXAR, WITNIT, KROOT, EPS, DELT, CLI00230 C COHHON / STANDARD BLOCK.) CLI00250 C COHHON / STANDARD BLOCK.) CLI00310 C COHHON / STANDARD BLOCK.) CLI00310 C COHHON / STANDARD BLOCK.) CLI00350 C COHHON / MISC/ MISC, HARRING BLOCK.) CLI00350 C COHHON / STANDARD BLOCK.) CLI00350 C COHHON / MISC/ MISC, HARRING BLOCK		DIMENSION VRIN(475), GEN(999), GREF(999), ALINK(99)	
1 IMDEX(37), NSYMB(36), DISS(35), PPASS(34), PCUM(33), CLI00140 2 PRIRCH(32), PCDND(31), PST(30), PROP(29), OPROP(28), CLI00150 3 OPRIOR(27), U(26), OU(25), WADJ(24), IDADJ(23), CLI00160 4 CIN(22), CTOT(21), OCIN(20), OBEN(19), VOLIN(18), CLI001/0 5 VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), CLI00180 6 PILE(12), XTHP(8), YTMP(7)) CLI00190 COHHON/CLUS/ JUNK(12), MARL,NTOP,NTBSZH,NWANT,LINK(14000) CLI00210 EQUIVALENCE (LR(1),LVRIN),(LR(2)),LKURT), CLI00220 1 (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) CLI00230 CC COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WITINIT,KROOT,EPS,DELT, 1 AMQ,DDCON,XOVFLO,XUMFLO,WADJIN,ELINTH,SEPTH,VFAC,AHN,SBLTH, CLI00250 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,AHOFAC, CLI00270 3 AHONIN,AHOHAX,AHORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIN,WDELSH, CLI00280 C (END OF STANDARD BLOCK.) C (END OF STANDARD BLOCK.) C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00310 C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00310 C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00330 C LI00330 C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00330 C C C C C C C C C C C C C C C C C C C		EQUIVALENCE (LINK(41), ALINK(41), GREF(8), GEN(7), VRIN(7))	
4 CIN(22), CTOT(21), OCIN(20), OBEN(19), VOLIN(18), CLI001/0 5 VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), CLI00180 6 PILE(12), XTHP(8), YTHP(7)) COHHON/CLUS/ JUNK(12),NARL,NTOP,NTBSZH,NWANT,LINK(14000) CLI00200 DINENSION HXAR(31),LR(3),LV(3) CLI00210 EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), CLI00220 1 (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) CLI00220 COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT, CLI00240 COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT, CLI00250 1 AMQ,ODCOM,XOVFLO,XUNFLO,WADJIN,ELIHTH,SEPTH,VFAC,AMM,SBLTH, CLI00260 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,ANDFAC, CLI00270 3 AMOMIN,AHOMAX,AHORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIM,WDELSH, CLI00270 C (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00330 COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00330 COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00340 CLI00330 COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAH,IPT,NOFLD, SYH(61), CLI00370 1 LNCAT, PRNT(4), KLBC, PRTNE, PROUT, TOTPIX, CLI00390 3, HAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLI00400 CLI00410 INTEGER TOTWRD,SYH,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT CLI00420 1,CLSNAH COHHON /MXLL/ HXLLWT, HXLLFN, RELPRP(200) CCOHHON /INITL/WTNEW,DEVINI,CHANIN COHHON /INITL/WTNEW,DEVINI,CHANIN		EQUIVALENCE (LINK(41), LSUBS(40), LSUPER(39), LIST(38),	
4 CIN(22), CTOT(21), OCIN(20), OBEN(19), VOLIN(18), CLI001/0 5 VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), CLI00180 6 PILE(12), XTHP(8), YTHP(7)) COHHON/CLUS/ JUNK(12),NARL,NTOP,NTBSZH,NWANT,LINK(14000) CLI00200 DINENSION HXAR(31),LR(3),LV(3) CLI00210 EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), CLI00220 1 (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) CLI00220 COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT, CLI00240 COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT, CLI00250 1 AMQ,ODCOM,XOVFLO,XUNFLO,WADJIN,ELIHTH,SEPTH,VFAC,AMM,SBLTH, CLI00260 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,ANDFAC, CLI00270 3 AMOMIN,AHOMAX,AHORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIM,WDELSH, CLI00270 C (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00330 COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00330 COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00340 CLI00330 COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAH,IPT,NOFLD, SYH(61), CLI00370 1 LNCAT, PRNT(4), KLBC, PRTNE, PROUT, TOTPIX, CLI00390 3, HAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLI00400 CLI00410 INTEGER TOTWRD,SYH,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT CLI00420 1,CLSNAH COHHON /MXLL/ HXLLWT, HXLLFN, RELPRP(200) CCOHHON /INITL/WTNEW,DEVINI,CHANIN COHHON /INITL/WTNEW,DEVINI,CHANIN		1 INDEX(37), NSYMB(36), DISS(35), PPASS(34), PCUM(33),	
4 CIN(22), CTOT(21), OCIN(20), OBEN(19), VOLIN(18), CLI001/0 5 VOLRT(17), DCON(16), SPFAC(15), PGRAT(14), PAVE(13), CLI00180 6 PILE(12), XTHP(8), YTHP(7)) COHHON/CLUS/ JUNK(12),NARL,NTOP,NTBSZH,NWANT,LINK(14000) CLI00200 DINENSION HXAR(31),LR(3),LV(3) CLI00210 EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), CLI00220 1 (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) CLI00220 COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT, CLI00240 COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WTINIT,KROOT,EPS,DELT, CLI00250 1 AMQ,ODCOM,XOVFLO,XUNFLO,WADJIN,ELIHTH,SEPTH,VFAC,AMM,SBLTH, CLI00260 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,ANDFAC, CLI00270 3 AMOMIN,AHOMAX,AHORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIM,WDELSH, CLI00270 C (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00310 CC (END OF STANDARD BLOCK.) CLI00330 COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00330 COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, CLI00340 CLI00330 COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAH,IPT,NOFLD, SYH(61), CLI00370 1 LNCAT, PRNT(4), KLBC, PRTNE, PROUT, TOTPIX, CLI00390 3, HAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLI00400 CLI00410 INTEGER TOTWRD,SYH,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT CLI00420 1,CLSNAH COHHON /MXLL/ HXLLWT, HXLLFN, RELPRP(200) CCOHHON /INITL/WTNEW,DEVINI,CHANIN COHHON /INITL/WTNEW,DEVINI,CHANIN		2 PRIRCH(32), PCOND(31), PST(30), PROP(29), OPROP(28),	
S		3 OPRIOR(27), W(26), OW(25), WADJ(24), IDADJ(23),	CL100160
6 PILE(12), XTHP(8), YTHP(7) COMHON/CLUS/ JUNK(12), NARL, NTDP, NTBSZH, NWANT, LINK(14000) DIHENSION MXAR(31), LR(3), LV(3) EQUIVALENCE (LR(1), LVRIN), (LR(2), LKURT), (LR(3), LOVAR), (LV(1), LSUM), (LV(2), LSKEW), (LV(3), LOSUM) COMHON /MISC/ HQ, HM, LR, LV, NINCLS, MXAR, WIINIT, KROOT, EPS, DELT, 1 AHQ, ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, ANM, SBLTH, 2 INDXVL, WFAC, NPTSO, PQRATH, SPMVTH, DWFAC, GRACTH, AMOFAC, 3 ANOMIN, AMOMAX, AMORAT, VOLLIH, BIAS, PJOIN, VRJOIN, WSIH, WDELSH, 4 BETTER, MODE C END OF STANDARD BLOCK.) COMHON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, 1 PACCEL(2), MACCEL(2), VACCEL(2) COMMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CLIO0340 CLIO0440		4 CIN(22), CTOT(21), OCIN(20), ODEN(19), VOLIN(18),	CLI001/0
COMMON/CLUS/ JUNK(12),NARL,NTOP,NTBSZH,NWANT,LINK(14000) DIMENSION MXAR(31),LR(3),LV(3) EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) COMMON /MISC/ MQ,MH,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT, 1 AMQ,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH, 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,AMOFAC, 3 AMOMIN,AMOMAX,AMORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIH,WDELSH, 4 BETTER, MODE (END OF STANDARD BLOCK.) COMMON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, 1 PACCEL(2),MACCEL(2),VACCEL(2) COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAH,IPT,NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, 2 SCRAM1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUFD 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLI00340		5	
DIMENSION MXAR(31),LR(3),LV(3) EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), CL100220 1 (LR(3),L0VAR),(LV(1),LSUM),(LV(2),LSKEW),(LV(3),LOSUM) CL100230 CL100240 CONMON /MISC/ HQ,HM,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT, AMQ,DDCON,XOVFLO,XUNFLO,WADJIN,ELIHTH,SEPTH,VFAC,ANM,SBLTH, L100260 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,AHOFAC, CL100270 3 AHOMIN,AHOMAX,AMORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIN,WDELSH, CL100290 C (END OF STANDARD BLOCK.) CDUBLE PRECISION XTEMP,YTEMP,ZTEMP,DURK,DURKD CC COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, PACCEL(2),MACCEL(2),VACCEL(2) COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYH(61), LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CL100370 CC COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYH(61), LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CL100370 CC COHHON/CLUSTR/ SEGIN, NUDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CL100410 CL100410 CL100410 CL100400			CLI00190
EQUIVALENCE (LR(1),LVRIN),(LR(2),LKURT), (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH) COHHON /MISC/ HQ,HH,LR,LV,NINCLS,HXAR,WIINIT,KROOT,EPS,DELT, AMG,ODCON,XOVFLO,XUMFLO,WADJIN,ELIHTH,SEPTH,VFAC,ANH,SBLTH, CL100240 INDXVL,WFAC,NPTSO,PORATH,SPHVTH,DWFAC,GRACTH,AHOFAC, AHOMIN,AHOHAX,AMORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIH,WDELSH, BETTER, HODE CEND OF STANDARD BLOCK.) COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, PACCEL(2),HACCEL(2),VACCEL(2) COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAH,IPT,NOFLD, SYH(61), LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, SCRAH1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUFD J, HAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IG1,NOCYCL, NCL CL100400 CL100440 CL100440 CL100440 CL100440			CLI00200
C COHHON /MISC/ HQ, HM, LR, LV, NINCLS, MXAR, WTINIT, KROOT, EPS, DELT, CL100240 COHHON /MISC/ HQ, HM, LR, LV, NINCLS, MXAR, WTINIT, KROOT, EPS, DELT, CL100250 1 AHQ, ODCON, XOVFLO, XUNFLO, WADJIN, ELIHTH, SEPTH, VFAC, ANM, SBLTH, CL100260 2 INDXVL, WFAC, NPTSO, PGRATH, SPHVTH, DWFAC, GRACTH, AHOFAC, CL100270 3 AMOHIN, AHOMAX, AHORAT, VOLLIH, BIAS, PJOIN, VRJOIN, WSIM, WDELSH, CL100280 4 BETTER, HODE CL100310 C END OF STANDARD BLOCK.) CL100310 C END OF STANDARD BLOCK.) CL100320 C COHHON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, CL100330 1 PACCEL(2), MACCEL(2), VACCEL(2) CL100350 C COHHON/CLUSTR/ IBEGIN, TOTWRD, CLSNAH, IPT, NOFLD, SYH(41), CL100340 1 LNCAT, PRNT(4), KLBC, PRTNE, PROUT, TOTPIX, CL100380 2 SCRAH1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD CL100390 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CL100410 C INTEGER TOTWRD, SYH, PRNT, PRTHE, PROUT, TOTPIX, SCRAH1, BUFPIX, BUFTOT CL100420 1 , CLSNAH CL100440 C COHHON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 C COHHON /INITL/WTNEW, DEVINI, CHANIN CL100470			CLI00210
CONHON /MISC/ HQ, HM, LR, LV, NINCLS, MXAR, WTINIT, KROOT, EPS, DELT, CL100250 1 AMG, ODCON, XOVFLO, XUNFLO, WADJIN, ELIMTH, SEPTH, WFAC, ANM, SBLTH, CL100260 2 INDXVL, WFAC, NPTSO, PGRATH, SPHVTH, DWFAC, GRACTH, AMGFAC, CL100270 3 AMOMIN, AMOMAX, AMORAT, VOLLIM, BIAS, PJOIN, VRJOIN, WSIN, WDELSH, CL100280 4 BETTER, HODE CL100370 C END OF STANDARD BLOCK.) CL100310 C DOUBLE PRECISION XTEMP, YTEMP, ZTEMP, DURK, DURKD CL100320 C COHHON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, CL100330 1 PACCEL(2), MACCEL(2), VACCEL(2) C COHHON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), CL100370 1 LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, CL100380 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD CL100390 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CL100400 C INTEGER TOTWRD, SYM, PRNT, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CL100420 1 , CLSNAM CL100440 C COHHON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 C COHHON /INITL/WTNEW, DEVINI, CHANIN CL100470			CL100220
COMMON /MISC/ MQ, MM,LR,LV,NINCLS,MXAR,WTINIT,KROOT,EPS,DELT, 1 AMQ,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH, 2 INDXVL,WFAC,NPTSO,PGRATH,SPMVTH,DWFAC,GRACTM,AMOFAC, 3 AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIM,WDELSH, 4 BETTER, MODE (END OF STANDARD BLOCK.) C CLIO0300 C CLIO0310 DOUBLE PRECISION XTEMP,YTEMP,ZTEMP,DURK,DURKD C COMMON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,WRKBND,WRKCHI, 1 PACCEL(2),MACCEL(2),VACCEL(2) C COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, C COMMON/CLUSTR/ BEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, C CLIO0330 2 SCRAM1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUFD 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLIO0410 C CLIO0410 C CLIO0430 C CLIO0430 C CLIO0430 C CLIO0440		1 (LR(3),LOVAR),(LV(1),LSUH),(LV(2),LSKEW),(LV(3),LOSUH)	CLI00230
1 AMG,ODCON,XOVFLO,XUNFLO,WADJIN,ELIMTH,SEPTH,VFAC,AMM,SBLTH, 2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,AMOFAC, 3 AMOMIN,AMOMAX,AMORAT,VOLLIM,BIAS,PJOIN,VRJOIN,WSIN,WDELSH, 4 BETTER, HODE C (END OF STANDARD BLOCK.) C CLIO0300 CLIO0310 DOUBLE PRECISION XTEMP,YTEMP,ZTEMP,DURK,DURKD C CLIO0330 CLIO0330 CCOMMON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, 1 PACCEL(2),HACCEL(2),VACCEL(2) C COMMON/CLUSTR/ IBEGIN,TOTWRD,CLSNAM,IPT,NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, 2 SCRAM1,BUFPIX,BUFTOT,NBUFSD,NBUMP,LBUFD 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLIO0430 C INTEGER TOTWRD,SYM,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT CLIO0430 C COMMON /HXLL/ HXLLWT, HXLLFN, RELPRP(200) C COMMON /INITL/WTNEW,DEVINI,CHANIN C COMMON /INITL/WTNEW,DEVINI,CHANIN	C		CLI00240
2 INDXVL,WFAC,NPTSO,PQRATH,SPHVTH,DWFAC,GRACTH,AHOFAC, 3 AHOHIN,AHOMAX,AMORAT,VOLLIH,BIAS,PJOIN,VRJOIN,WSIM,WDELSH, 4 BETTER, HODE C (END OF STANDARD BLOCK.) C CLI00300 C CLI00310 DOUBLE PRECISION XTEMP,YTEMP,ZTEMP,DURK,DURKD C CLI00320 CLI00330 C COHHON /STPAR/WAIT,CONLV,SKBND,SKCHI,TRBND,TRCHI,URKBND,URKCHI, 1 PACCEL(2),HACCEL(2),VACCEL(2) C CLI00350 C COHHON/CLUSTR/ IBEGIN,TOTWRD,CLSNAH,IPT,NOFLD, SYH(61), 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, 2 SCRAH1,BUFPIX,BUFTOT,NBUFSD,NDUMP,LBUFD 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1,NOCYCL, NCL C LI00410 C LI00410 C LI00420 1 ,CLSNAH C COHHON /HXLL/ HXLLWT, HXLLFN, RELPRP(200) C COHHON /INITL/WTNEW,DEVINI,CHANIN C COHHON /INITL/WTNEW,DEVINI,CHANIN		CONHON /MISC/ HQ, HH, LR, LV, NINCLS, MXAR, WTINIT, KROOT, EPS, DELT,	CL100250
A AMOMIN, ANOMAX, AMÓRAT, VOLLIM, BIÁS, PJOÍN, VRJOÍN, WSIN, WDELSH, A BETTER, HODE C (END OF STANDARD BLOCK.) C CLIO0300 CLIO0310 DOUBLE PRECISION XTEMP, YTEMP, ZTEMP, DURK, DURKD C CLIO0320 CLIO0330 COHNON / STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, 1 PACCEL(2), HACCEL(2), VACCEL(2) C COHNON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYH(61), 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, HAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1, NOCYCL, NCL C INTEGER TOTWRD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT C CLIO0440 C COHNON / MXLL/ HXLLWT, HXLLFN, RELPRP(200) C COHNON / INITL/WTNEW, DEVINI, CHANIN C CLIO0470		1 AMO, ODCON, XOVFLO, XUNFLO, WADJIN, ELINTH, SEPTH, VFAC, ANN, SBLTH,	CLI00260
4 BETTER, MODE C (END OF STANDARD BLOCK.) C CL100300 C CL100310 DOUBLE PRECISION XTEMP, YTEMP, ZTEMP, DURK, DURKD C CL100320 C COHMON / STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, CL100340 1 PACCEL(2), MACCEL(2), VACCEL(2) C COHMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), CL100370 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CL100380 2 SCRAMI, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD C CL100400 C CL100400 C CL100410 1, CLSNAM C CL100440 C COHMON / MXLL/ HXLLWT, MXLLFN, RELPRP(200) C COHMON / INITL/WTNEW, DEVINI, CHANIN C CL100470		2 INDXVL, WFAC, NPTSD, PQRATH, SPHVTH, DWFAC, GRACTH, ANOFAC,	CLI00270
C (END OF STANDARD BLOCK.) C DOUBLE PRECISION XTEMP, YTEMP, ZTEMP, DURK, DURKD C COHMON / STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, PACCEL(2), HACCEL(2), VACCEL(2) C COHMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD ANABF, AREA, NWDS, NWDRS, NPTS, LBUF, IG1, NOCYCL, NCL CL100410 C CL100410 CL100420 CL100430 CL100430 CL100430 CL100440 CL100440 CL100440 CL100440 CL100440 CL100440 CL100440 CL100440		3 ANONIN, ANONAX, ANORAT, VOLLIH, BIAS, PJOIN, VRJOIN, WSIH, WDELSH,	CL100280
CLI00310 CDUBLE PRECISION XTEMP, YTEMP, ZTEMP, DURK, DURKD CLI00320 CLI00330 COMMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, CLI00340 1 PACCEL(2), MACCEL(2), VACCEL(2) COMMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), CLI00350 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CLI00380 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CLI00400 C INTEGER TOTWRD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CLI00420 1, CLSNAM C COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) C COMMON /INITL/WTNEW, DEVINI, CHANIN CLI00470			CLI00290
CLIO0320 COMMON /STPAR/WAIT, CONLY, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, PACCEL(2), MACCEL(2), VACCEL(2) COMMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, SCRAMI, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD J, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CIO0410 INTEGER TOTWRD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAMI, BUFPIX, BUFTOT CLIO0430 CLIO0430 CCCCOMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) COMMON /INITL/WTNEW, DEVINI, CHANIN CLIO0470 CLIO0440 CLIO0440	C	(END OF STANDARD BLOCK.)	CL100300
CUMMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, 1 PACCEL(2), MACCEL(2), VACCEL(2) CUMMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CLI00410 INTEGER TOTWRD, SYM, PRNT, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CLI00420 1, CLSNAM CUMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) COMMON /INITL/WTNEW, DEVINI, CHANIN CLI00470	C		CLI00310
COMMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI, 1 PACCEL(2), MACCEL(2), VACCEL(2) C COMMON/CLUSTR/ IBEGIN, TOTURD, CLSNAM, IPT, NOFLD, SYM(61), 1 LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IQ1, NOCYCL, NCL C CLIO0410 INTEGER TOTURD, SYM, PRNT, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CLIO0430 CLIO0440 C COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) C COMMON /INITL/WINEW, DEVINI, CHANIN CLIO0470		DOUBLE PRECISION XTEMP, YTEMP, ZTEMP, DURK, DURKD	CL100320
C COMMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61), CLI00350 LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, CLI00380 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CLI00400 C INTEGER TOTWRD, SYM, PRNT, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CLI00420 1, CLSNAM CLI00430 C COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) C COMMON /INITL/WTNEW, DEVINI, CHANIN CLI00470	E		CL100330
CL100360 COMMON/CLUSTR/ IBEGIN, TOTURD, CLSNAM, IPT, NOFLD, SYM(61), CL100370 1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CL100380 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CL100400 C INTEGER TOTURD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CL100420 1, CLSNAM CL100430 C COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 C COMMON /INITL/WTNEW, DEVINI, CHANIN CL100470		COMMON /STPAR/WAIT, CONLV, SKBND, SKCHI, TRBND, TRCHI, URKBND, URKCHI,	CLI00340
COMMON/CLUSTR/ IBEGIN, TOTURD, CLSNAM, IPT, NOFLD, SYM(61), LNCAT, PRNT(4), KLBC, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD AMABF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1, NOCYCL, NCL CLIO0400 CLIO0410 INTEGER TOTURD, SYM, PRNT, PRTME, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CLIO0430 CLIO0430 CLIO0440 CLIO0450 CLIO0450 CLIO0450 CLIO0450 CLIO0470			CL100350
1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CL100380 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD CL100390 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IG1, NOCYCL, NCL CL100400 C INTEGER TOTWRD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CL100420 1, CLSNAM CL100430 C COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 C COMMON /INITL/WTNEW, DEVINI, CHANIN CL100470	C		CL100360
1 LNCAT, PRNT(4), KLBC, PRTHE, PROUT, TOTPIX, CL100380 2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD CL100390 3, MAXBF, AREA, NWDS, NWDRS, NPTS, LBUF, IG1, NOCYCL, NCL CL100400 C INTEGER TOTWRD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CL100420 1, CLSNAM CL100430 C COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 C COMMON /INITL/WTNEW, DEVINI, CHANIN CL100470		COMMON/CLUSTR/ IBEGIN, TOTWRD, CLSNAM, IPT, NOFLD, SYM(61) ,	CLI00370
2 SCRAM1, BUFPIX, BUFTOT, NBUFSD, NDUMP, LBUFD 3, MAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IG1, NOCYCL, NCL CLIO0400 CLIO0410 INTEGER TOTURD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT CLIO0430 CLIO0430 CLIO0440 CLIO0450 CLIO0450 CLIO0450 CLIO0460 CLIO0470			CLI00380
3, MAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1,NOCYCL, NCL CLIO0400 CLIO0410 INTEGER TOTURD,SYM,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT CLIO0420 1,CLSNAM CLIO0430 CLIO0440 COMMON /MXLL/ MXLLUT, MXLLFN, RELPRP(200) CLIO0450 CLIO0450 CLIO0460 CLIO0470			CL100390
CLIO0410 INTEGER TOTURD,SYM,PRNT,PRTHE,PROUT,TOTPIX,SCRAM1,BUFPIX,BUFTOT CLIO0420 1,CLSNAM CLIO0430 CLIO0440 CLIO0440 CLIO0450 CLIO0450 CLIO0460 CLIO0460 CLIO0470		3, NAXBF, AREA, NUDS, NUDRS, NPTS, LBUF, IQ1,NOCYCL, NCL	CLI00400
1 , CLSNAH CL100430 C COMMON / MXLL/ HXLLWT, HXLLFN, RELPRP(200) CL100450 C COMMON / INITL/WTNEW, DEVINI, CHANIN CL100470	C		CLI00410
1 , CLSNAH CL100430 C COMMON / MXLL/ HXLLWT, HXLLFN, RELPRP(200) CL100450 C COMMON / INITL/WTNEW, DEVINI, CHANIN CL100470		INTEGER TOTWRD, SYM, PRNT, PRTHE, PROUT, TOTPIX, SCRAM1, BUFPIX, BUFTOT	CLI00420
COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 CC COMMON /INITL/WTNEW, DEVINI, CHANIN CL100470			CLI00430
COMMON /MXLL/ MXLLWT, MXLLFN, RELPRP(200) CL100450 C COMMON /INITL/WTNEW, DEVINI, CHANIN CL100470	C	•	CL100440
C COMMON /INITL/WTNEW, DEVINI, CHANIN CLIO0450		CONNON /MXLL/ HXLLWT, HXLLFN, RELPRP(200)	
COMMON /INITL/WINEW, DEVINI, CHANIN CLIO0470	C	•	CL100460
		COMMON /INITL/WTNEW,DEVINI,CHANIN	CL100470
		CHIVAL(DF)=DF*(1222/DF+CONLV+SQRT(.222/DF))**3	CL100480

	OF POOR QUALITY	
C	INITIALIZS ADDRESS	CLI00490
C		CLI00500
	CALL DATFIX	CL100510
C		CLI00520
	ANC=NC	CL100530
	MQS=MQ*MQ	CLI00540
C		CL100550
C	DEFINE VALUE OF SEPTH IN TERMS OF CHI SQUARE VALUE	CL100560
_	DEFINE VALUE OF SEPTH IN TERMS OF CHI SQUARE VALUE DFT = ANQ + 1	CL100570
	SEPTH = (CHIVAL(DFT))/2	CL100580
C		CLI00590
Č	WE FIRST SET UP VARIOUS INDEX ARRAYS FOR A PARTICULAR	CL100600
Ç	NUMBER OF CHANNELS HQ.	CLI00610
C		CLI00620
•		CL100630
		CLI00640
	HXAR(I)=HH	CL100650
	10 MH=MM+I	CL100660
	10 HH=HH+I HH=HXAR(HQ+1)	CL100670
		CL100680
^	ANN=NN	CL100690
C	SET UP TABLES FOR THE XP FUNCTION CALL XPREP	CL100870
~		
	NOW HE SET UP THE ORIGIN VECTORS, LR AND LV, OF THE VARIOUS ARRAYS	
C	AND VECTORS IN A CLUSTER NODE.	CL100720
_	NINCLS=1	· CLI00730
Ü	***** THIS CONSTANT MUST BE SET TO THE NUMBER OF ARRAYS *****	CL100740
	DO 21 I=1,3	CL100750
	LR(I)=NINCLS	CL100760
	21 NINCLS=NINCLS+NN	CL100770
	DO 22 I=1,3	CL100780
	LV(I)=NINCLS	CLI00790
	22 HINGLS=HINGLS+HQ	CT100800
	NSCALS = 35	CL100810
_	NINCLS=NINCLS+NSCALS-1	CL100820
C	WE HUST ALSO SET UP SOME THRESHOLDS FOR USE BY THE STATISTICAL	
C	SYSTEM.	CL100840
	SKCHI=(AMQ+2.)*(AMQ+4.)*CHIVAL(AMQ)	CLI00850
	URKCHI=AHQ*(AHQ+4.)*(AHQ+6.)/(AHQ999)*CHIVAL(AHH-1.)	CF100890
	TRCHI=CONLV*CONLV*(AHQ*(AHQ+2.)*(AHQ+3.)*8.)	CL100870
C	WE NOW CREATE THE HEAD NODE OF THE CLUSTER TREE. THIS IS NOT	CL100880
C	AN ACTUAL CLUSTER, AND DOES NOT HAVE STORAGE FOR ANY	CL100890
C	OF THE STATISTICAL ARRAYS.	CL100900
	NPTSO=0	CLI00910
	KROT=HORSTR(NSCALS)	CLI00920
C	NAKE FIRST NODE START AT AN ODD NUMBER	CL100930
	IF (HOD(NTOP,2) .NE. 1) HTOP = NTOP + 1	CLI00940
	LINK(KROT)=-262139	CL100950
	LSUPER(KROT)=-262142	CLI00960

	IDADJ(KROT)=999999	CLI00970
	INDEX(KROT)=-0	CL100980
	SPFAC(KROT)=99999.	CLI00990
	W(KROT)=WTINIT	CLI01000
	OW(KROT)=W(KROT)	CLI01010
	PQRAT(KROT)=0.	CLI01020
	PROP(KROT)=1.	CLI01030
	OPROP(KROT)=1.	CL101040
	CIN(KROT)=W(KROT)	CLI01050
	OCIN(KROT)=CIN(KROT)	CLI01060
	CTOT(KROT)=0.	CL101070
	ODEN(KROT)=W(KROT)	CLI01080
	PRIRCH(KROT)=1.	CLI01090
С	NEXT THE INITIAL NODE IS SET UP, TOGETHER WITH SOME CONTROL THRESHOL	DCL101100
•	ODCON =60.	CLI01110
	PROPI = 1.	CLI01120
	INDXVL = 0	CLI01130
	57 KFIR = NEWCLS(KROT, PROPI, WTINIT)	CLI01140
	LSUBS(KROT) =KFIR	CLI01150
	LINK(KFIR)=0	CLI01160
	DEV2UT =DEVINI*DEVINI*WTINIT	CLI01170
	LA=HORSTR(HQS)	CLI01180
	DO 52 I=1.HQ	CLI01170
	GREF(KFIR+LSUM+I)=WTINIT*CHANIN	CLI01200
	DO 52 J=1, NO	CLI01210
	IJ=J-1+(I-1)*HQ+LA	CLI01220
	ALINK(IJ) =0.	CLI01230
	IF(I.EQ.J) ALINK(IJ)=DEV2WT	CLI01240
	52 CONTINUE	CL101250
C		CLI01260
_	CALL ZEROCL(KFIR, ALINK(LA), GREF(KFIR+LSUN+1), GREF(KFIR+LSKEU+1),	CLI01270
		CLI01280
	CALL FREE(LA, MQS)	CLI01290
C	SET SWITCHES FOR MAX LIKLIHOOD LABELING	CL101300
_	HXLLUT = 0	CLI01310
	WADJ(KFIR) =WADJIN	CLI01320
	MXLLFN = 23	CLI01330
	PRINT 273, MG, CONLV, TRCHI, SKCHI, URKCHI	CLI01340
	273 FORMAT ('1 CONFIDENCE LEVELS', 14, ' CHANNELS', F8.4, ' CHISQUARES',	
	1 3F12.2)	CL101360
C	1 41 1414	CLI01370
C		CLI01380
C	INITIALIZE ADDRESS:	CLI01390
Č		CL101400
_	CALL STATAD(GEN(LSUM),GEN(LSKEW),GEN(LKURT),	CLI01410
	1 GEN(LOSUM),GEN(LOVAR))	CL101420
	RETURN	CLI01430
	END	CLI01440
	•	